ARMY RESEARCH LABORATORY

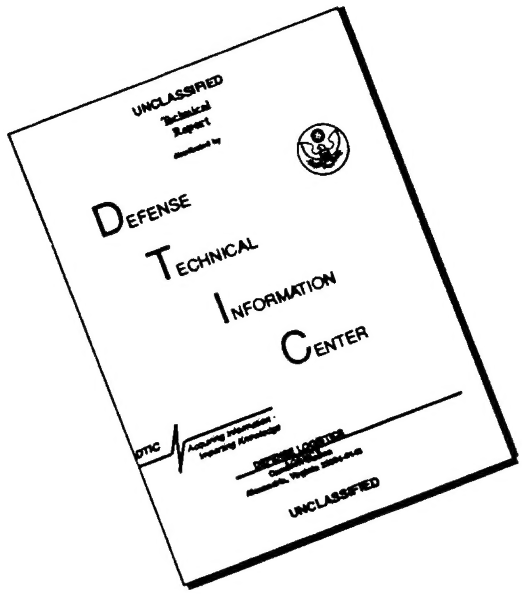
Targeting Regions of Chaotic Attractors Using Small Perturbation Control of Symbolic Dynamics

by Chance M. Glenn and Scott Hayes

ARL-TR-903

May 1996

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1996

3. REPORT TYPE AND DATES COVERED
Final, September 1995

4. TITLE AND SUBTITLE
Targeting Regions of Chaotic Attractors Using Small Perturbation Control of Symbolic Dynamics

5. FUNDING NUMBERS
PE: S18129

6. AUTHOR(S)
Chance M. Glenn and Scott Hayes

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn AMSRL-IS-TA
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-903

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
AMS code: 622120.H16
ARL PR: 6FEJ10

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

It has been shown both theoretically and experimentally that it is possible to control the *symbolic dynamics* of a chaotic system using small perturbations. We show a method of very rapidly targeting a region of width ε near a desired state point on a one-dimensional map using control of symbolic dynamics; this method demonstrates that controlling symbolic dynamics provides an efficient algorithm for targeting.

14. SUBJECT TERMS
Chaos, targeting, symbolic dynamics, dynamical systems, maps

15. NUMBER OF PAGES
29

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Contents

Figures

Tables

1. Introduction

The presence of a type of dynamical behavior, termed *chaotic*, has been observed experimentally, analytically, and numerically in many diverse nonlinear systems [1,2]. Researchers have moved beyond the point of categorizing this behavior as unwanted, although at first observation it may appear to be noisy and unpredictable. Since the fundamental assertion by Ott, Grebogi, and Yorke (OGY) that chaotic dynamics can be controlled by the use of small perturbations [3], a great deal of research has been conducted that not only experimentally verifies the assertion but also proposes useful applications of these ideas. Much of the research has been centered around controlling and stabilizing the unstable periodic orbits [4] that lie within chaotic orbits. Recent theoretical and experimental work shows that the *symbolic dynamics* [5] of a chaotic system can be controlled as well [6,7].

In many driven nonlinear systems, changing (e.g., increasing) a drive parameter can be an impetus to chaotic behavior. For example, a modified Van der Pol equation used by Ueda to model a negative resistor oscillator [1] has the form $\ddot{x} + (x^2 - 1)\dot{x} + x^3 = B \cos \omega t$. This oscillator experiences steady-state, periodic, quasi-periodic, and chaotic motion depending upon the parameters B and ω. Many experimenters have observed chaotic behavior emerge as drive parameters change [8–10]; those regions of behavior have usually been considered undesirable. With control-of-chaos techniques, however, it is possible to stabilize any of the dense set of periodic orbits embedded in regions of chaotic behavior, thus creating the possibility of more robust operation in the chaotic behavior region.

Present control-of-chaos techniques based upon the OGY assertion must wait for the ergodic wanderings of the chaotic trajectories to come near a desired periodic point. This is necessary so that (a) there is a linearizable region near the periodic point for which to calculate control perturbations and (b) the control perturbations are not so large as to significantly deform the topology of the state-space attractor. A state-space attractor is an object in the state space to which the trajectories tend [4].

For some systems that may be operating near *crisis* [4,11], it may not be desirable to wait for the trajectories to wander near a desired periodic point (control point). In this case, it is necessary to use the inherent sensitivity to small changes to guide the trajectories near a control point as rapidly as possible. This is the notion of targeting first formally introduced in a chaotic system by Shinbrot et al [12]. In this report, we discuss a particular method of targeting using small perturbation control of symbolic dynamics.

2. Discrete-Time Maps

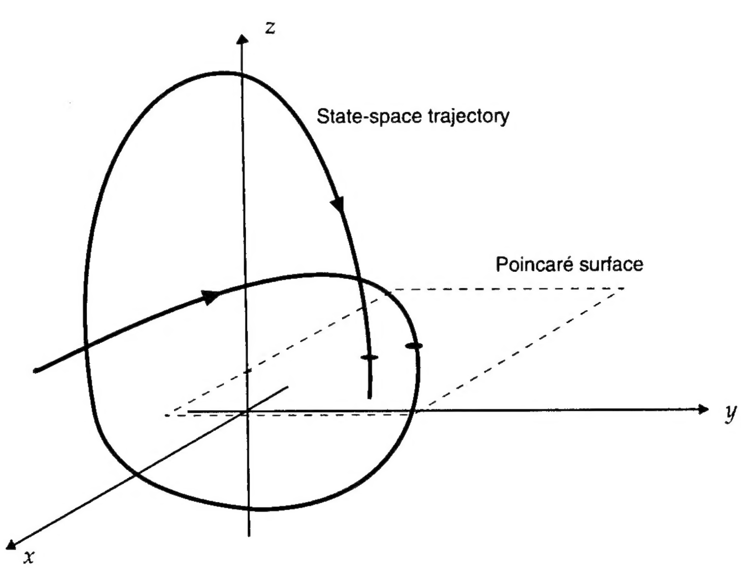
A large class of continuous-time nonlinear systems that behave chaotically can be described mathematically by a three-dimensional system of nonlinear ordinary differential equations [1]. The solutions to these systems form a continuous-time flow in three-dimensional space. This space has been termed a *state space,* simply because at any time $t = t_s$ the solutions $x(t_s)$, $y(t_s)$, $z(t_s)$ describe a point (state point) in the space that also describes the *state* of the system. This continuous-time system can be reduced to a discrete-time map by judicious placement of a surface that cuts through the flow of state-space trajectories. The points that are produced as the trajectories pass through the surface create a two-dimensional discrete-time mapping of state points called the *Poincaré surface of section* [4]. Figure 1 shows a continuous-time flow discretized by a Poincaré surface of section placed at constant z, thus producing a two-dimensional map described by $[x_{n+1}, y_{n+1}] = \mathbf{F}(x_n, y_n)$. Poincaré surfaces of section for a large class of highly *dissipative* nonlinear systems can produce approximately one-dimensional discrete-time surfaces [13,14]. If we focus on one of the discrete-time state points, say x_n, then we can form a mapping based upon its sequential surface crossings. This map is commonly called a *return map,* which has the form $x_{n+1} = f(x_n)$, and will also be approximately one-dimensional for a dissipative system. We focus on this particular case and show how the formalism can be generalized for higher dimensional maps.

A paradigm for the study of one-dimensional chaotic maps is the logistic map. It is a single-maximum, noninvertible map described by the expression $x_{n+1} = \rho x_n(1 - x_n)$. We use the following noise-perturbed map:

$$x_{n+1} = f(x_n) + \xi_n = \rho x_n (1 - x_n) + \xi_n , \tag{1}$$

where ρ is a parameter that governs the global behavior of the map, and x_n is a small amount of uniformly distributed, zero-mean noise with a peak-to-peak amplitude of 10^{-12} that we added to avoid the *round-off induced*

Figure 1. A three-dimensional chaotic attractor demonstrating how state-space trajectories pierce Poincaré surface of section.



periodicity inherent in computer simulations [15]. Figure 2 shows the map created by successive iteration of equation (1) for $\rho = 3.97$. The points that are generated by successive iterations of the map can be referred to as the attracting set.

We can find the boundaries of the map by first finding the state point x_p that maps to the maximum edge of the attractor as a function of ρ. At the peak, we know that

$$\left.\frac{\partial f}{\partial x}\right|_{x = x_p} = \rho - 2\rho x = 0 \ , \tag{2}$$

which has the solution $x = x_p = 1/2$. Substituting this value into the expression for the map gives

$$x_{\max} = f(x = x_p) = \rho x_p (1 - x_p) = \frac{\rho}{4} \ , \tag{3}$$

which is the maximum height of the map. Successive iterations of the map defined by equation (1) produce a sequence of points x_n, which wander ergodically [4] over an attracting set defined by a minimum and maximum edge $[x_{\min}, x_{\max}]$ ($x_{\max}$ not only defines the peak of the map, but the maximum boundary of the attracting set as well). We determine the minimum boundary of the attracting set, $x_{\min}$, by

$$x_{\min} = f(x = x_{\max}) = \rho x_{\max}(1 - x_{\max}) = \frac{1}{16}\rho^2(4 - \rho) \ . \tag{4}$$

This value is important in determining constraints on the *grammar* of the map, which we discuss in section 6. Also note that we are considering only state points that lie in the unit interval on the real axis. This is true for $0 < \rho < 4$.

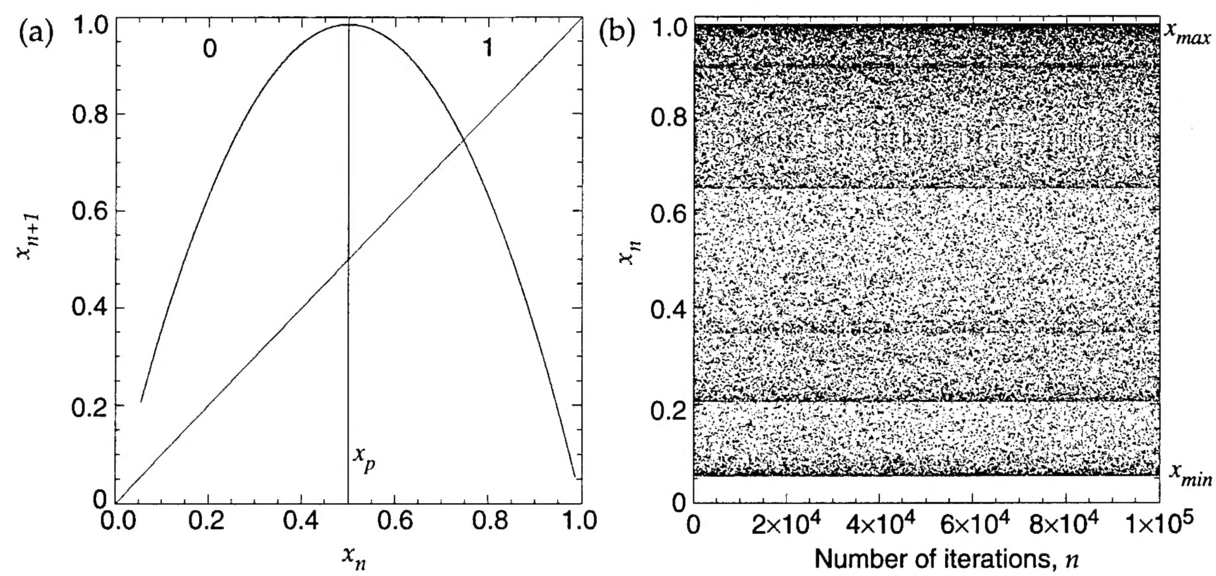


Figure 2. (a) Return map and (b) sequential output (attracting set) of logistic map for $\rho = 3.97$ in presence of uniformly distributed, zero-mean noise of maximum amplitude 10^{-12}.

3. Symbolic Dynamics Representation of 1-D Chaotic Maps

If we partition the unit interval into disjoint subsets called "atoms" or "partition elements" so that a state point falling into a given partition element is assigned a particular symbol, then we form a *symbolic* representation of the dynamics. For a binary symbolic system, we choose the peak of the map ($x = 1/2$) as the partition boundary. The choice of this partition is not arbitrary. For the logistic map, this point is known as the *generating partition* because it and all its iterates and pre-iterates divide the state space arbitrarily finely [16]. In other words, each state point has its own unique symbol sequence assigned to it. As x_n is iterated forward, we can assign a "0" to a symbol b_n when a state-space point lands on the left side of the partition, and assign a "1" to the symbol b_n when a state-space point lands on the right side (see fig. 2a). The bits are shifted into the right side of an N-bit register B_n, and the left-most bit is thrown out (see fig. 3). For example, a 16-bit symbol sequence describing a trajectory that continuously passes through the *fixed point* would be B_n = 1111111111111111 (the fixed point, or period-1 point, is the state point for which $x_{n+1} = x_n$). A period-2 sequence could be B_n = 1010101010101010 or B_n = 0101010101010101. Now we want to know what symbol sequence we can expect, given that we have just passed through a particular point in state space. In other words, we ask what state point *sources* a given symbol sequence. In order to do this we create a *binary coding function* [6,7]. (From this point on, we work with 16-bit symbol sequences.) We define the number k as the value of the symbol sequence simply by

$$k = \sum_{n=0}^{N-1} 2^n b_{N-n} \ . \tag{5}$$

Now setting $N = 16$ and iterating the map 16 times, we collect the set of state points $[x_0, x_1, \dots x_{15}]$ and the corresponding symbol sequence $B = [b_0, b_1, \dots b_{15}]$. We say that the first state point x_0 *evolves* the symbol sequence B, which has value k. (In the following, for simplicity of notation, we refer to k interchangeably as the value of the symbol sequence and the sequence itself.) Continuous iteration of the map from various source locations will form a state-point/symbol-sequence relationship for the map called the inverse coding function:

$$s[k] = x_k \ . \tag{6}$$

That is, the state point x_k sources the symbol k.

Forming this inverse coding function is part of what we call the *characterization* of the system. Figure 4 shows a 16-bit inverse coding function for

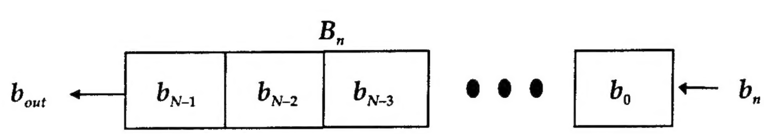


Figure 3. Bit-shifting diagram for a finite-symbol sequence register.

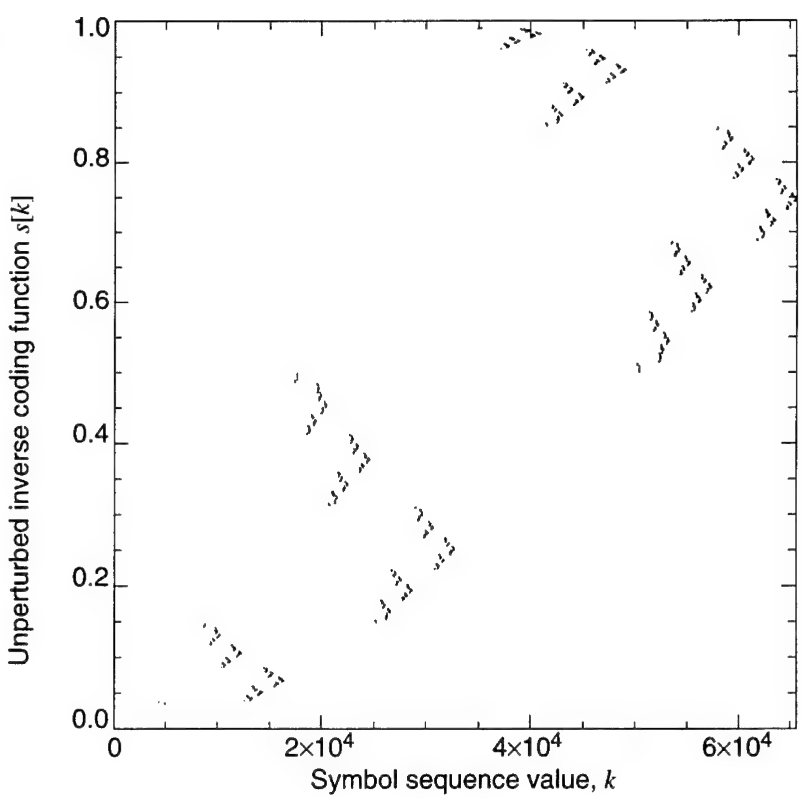
the logistic map when $\rho = 3.97$. Despite the unusual topology of the "function," every state point sources a unique symbol sequence. We will see that this structure is based primarily upon the ordering of the symbol sequences across the state space. Symbol sequences are evolved from finite regions, Δ_k, of the state space that have widths on the order of 2^{-16}.* A simple way of determining a single source point for a given symbol sequence k is by averaging over the state points producing that symbol sequence (all state points that fall into a subregion Δ_k will produce the sequence k). Namely,

$$S[k] \cong \bar{x}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} x_{k,j} \ , \tag{7}$$

where N_k is the number of observed state points that source the given symbol sequence k during data collection, and $x_{k,j}$ is the jth state point that falls into the subregion Δ_k. We can find a sequential expression for the partition refinements if we first realize that the two-bit partition x_{p2} occurs where x_{p1} (one-bit partition, x_p) intersects the $x_n = x_{n+1}$ line. In other words, we solve

$$f(x_{p2}) = rx_{p2}\,(1 - x_{p2}) = x_{p1} \ , \tag{8}$$

Figure 4. 16-bit coding function, $s[k]$, for the logistic map at ρ = 3.97 in presence of 10^{-12} noise.



* *Inverse images, that is, "backward" iterations of the partition {[9, 1/2), [1/2, 1)}, and successive intersections, form refinements of the partition with 2N pieces of average width $\bar{\Delta}_k = 2^{-N}$ at the Nth stage of refinement.*

which is a quadratic expression from which we can derive the general expression for the refinements,*

$$x_{p_{j+1}} = \Phi\left(\rho, x_{p_j}\right) = \frac{\rho \pm \sqrt{\rho^2 - 4\rho x_{p_j}}}{2\rho}\ , j = 1, 2, \ldots \tag{9}$$

If there is a complex solution to equation (9) for a given partition, then that finite-length sequence is not allowed; thus the grammar is constrained. The ordering of the symbol sequences, across the map from left to right, conforms to a *Gray* ordering [17]. Table 1 shows three-bit sequences ordered according to the standard binary "counting" order, and ordered according to a "Gray code."

Figure 5 shows a plot of the binary value of sequences in a Gray ordering versus their position in the list. Note the similarity to the coding function in figure 4. If the partitions are known and the ordering is known, then everything necessary to characterize the system is known. In other words, given the expression for the map, a complete N-bit coding function can be *calculated.* This would serve to significantly reduce the characterization time. For systems where the map is not known explicitly (e.g., experimental systems, complex models), yet the system is dissipative, curve-fitting techniques can be used to fit the return map, and root-finding methods can be used to determine the refinement boundaries. Figure 6 shows a four-bit refinement completely determined by this method.

Figure 7 shows the probability mass function for 16-bit symbol sequences. It is apparent that certain sequences have a higher probability of occurrence than others. There are also constraints on the system that disallow certain sequences.

Table 1. Three-bit Gray ordering example.

k	Binary	Gray ordering
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

The partition is a collection of intervals $P = \{P_i\}$. Refinement is $P^n = P \vee f^{-1} P \vee .. \vee f^{1-n}P$ at step n, where $P \vee Q \triangleq \{P_i \cap Q_i \mid \forall\ P_i \in P \text{ and } Q_i \in Q\}$.

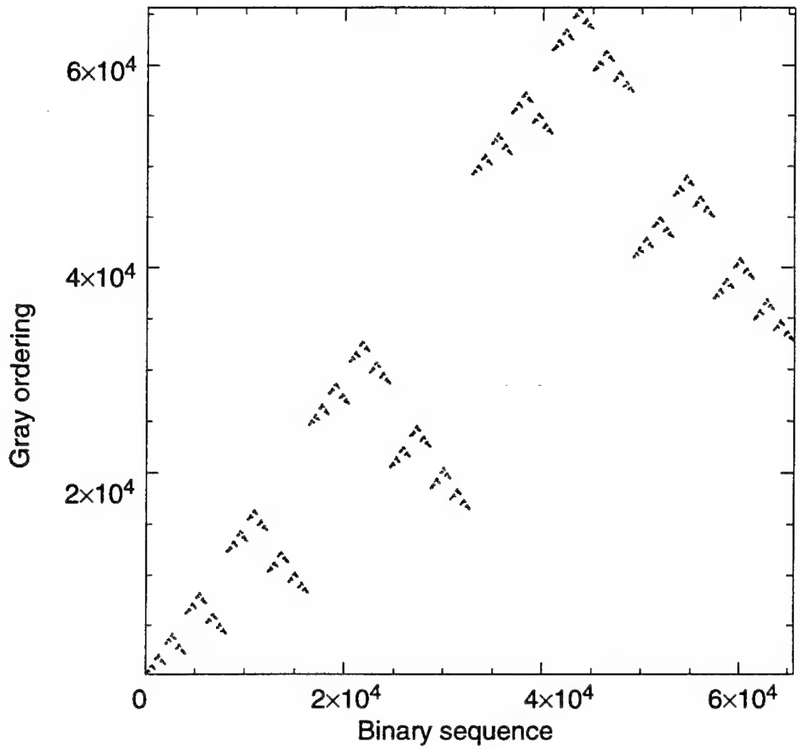


Figure 5. 16-bit binary-to-Gray ordering.

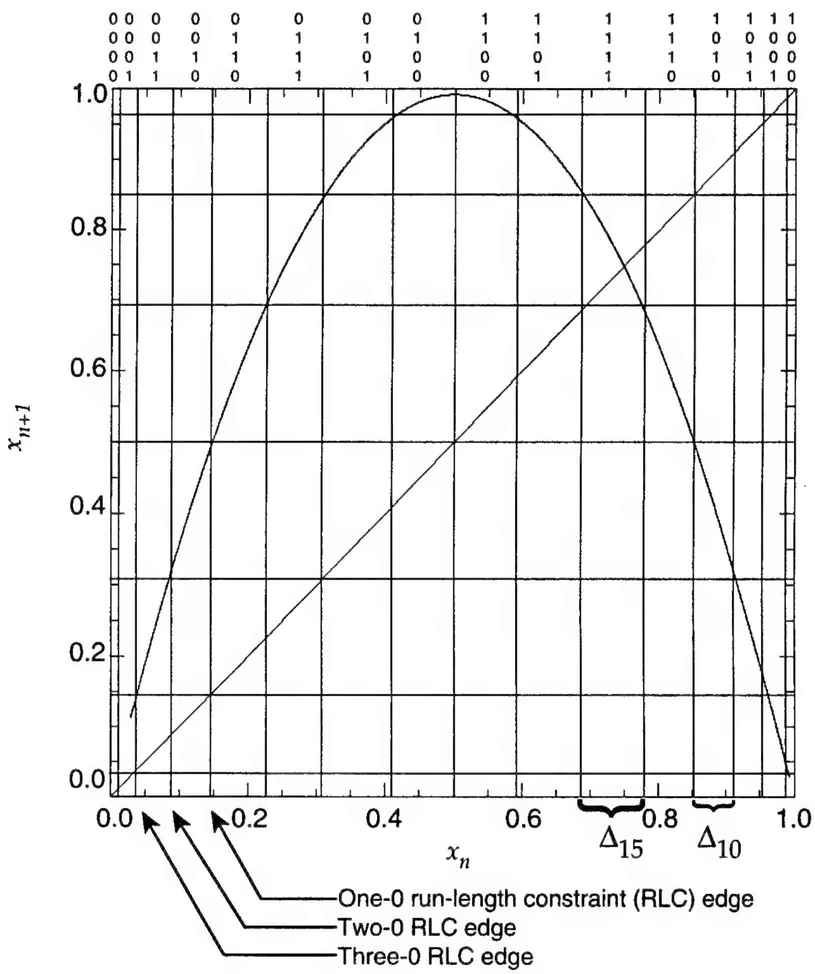
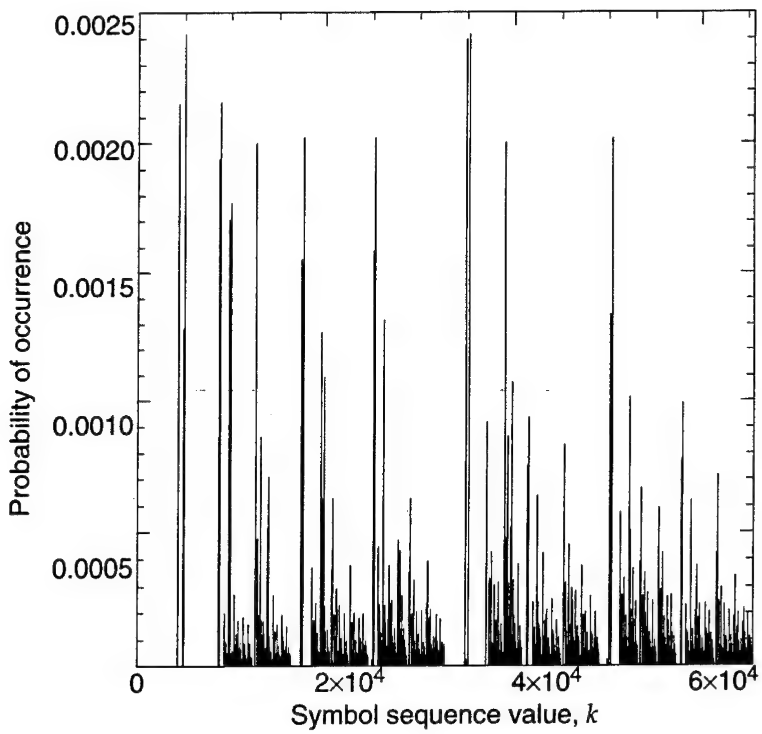


Figure 6. Four-bit partition diagram for the logistic map.

Figure 7. Probability of occurrence for 16-bit symbol sequences.



4. Characterization Under the Influence of Reference Perturbations

We now determine what happens to the system, namely the inverse coding function (which is a complete description of the symbolic dynamics), when influenced by small perturbations. In this phase of the characterization, we form a new inverse coding function, $q[k]$, which describes the dynamics of the system under the influence of reference perturbations. As the state trajectory generates the first symbol, we add a small reference perturbation, p_{ref}, to the parameter ρ. The map becomes

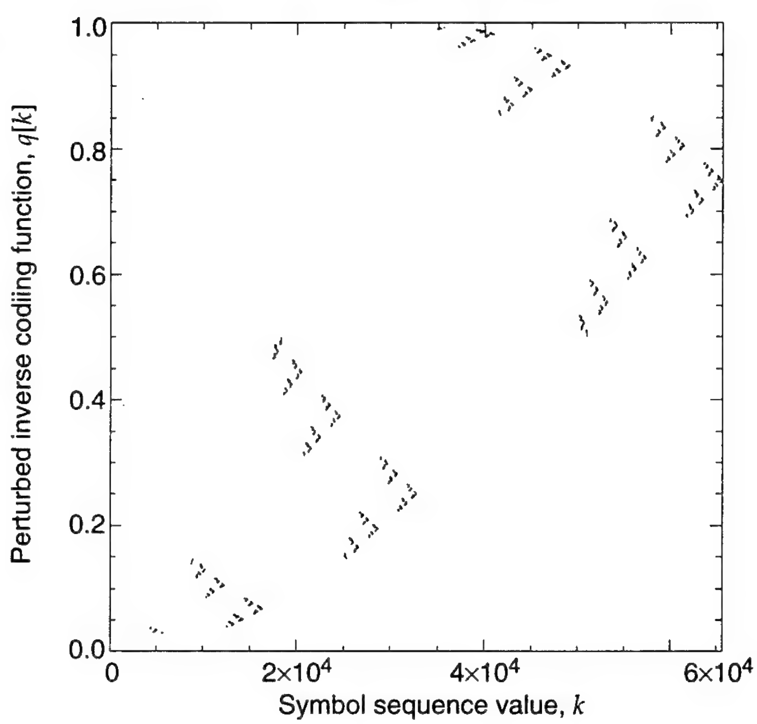
$$x_{n+1} = \left(\rho + p_{ref}\right)x_n\left(1 - x_n\right) + \xi_n \ . \tag{10}$$

The next 15 symbols are observed but the parameter is set back to r. We thus see the effect of a small perturbation on the evolving sequence. We now have a new inverse coding function that tells us what state point sources each allowed sequence *if a reference perturbation is present*. Figure 8 shows that $q[k]$ is qualitatively similar to $s[k]$, shown in figure 4.

The reference perturbation actually changes the symbolic *future* of the state-space trajectory (the current symbol is not affected as the perturbation is applied). Consequently, we can view this as changing the *effective* sourcing state point for the symbol sequence k. As a result of our characterization phases, we know that for a reference perturbation, p_{ref}, there will be a measurable difference between the state point that sources the symbol k (i.e., $q[k]$) and the state point that sources the symbol k under zero perturbation (i.e., $s[k]$). In other words,

$$\delta x_k = s[k] - q[k] \ . \tag{11}$$

Figure 8. 16-bit coding function, $q[k]$, under influence of 10^{-2} uniform reference perturbations applied every 16 iterations.



Using this result we form a proportionality constant for each symbol sequence:

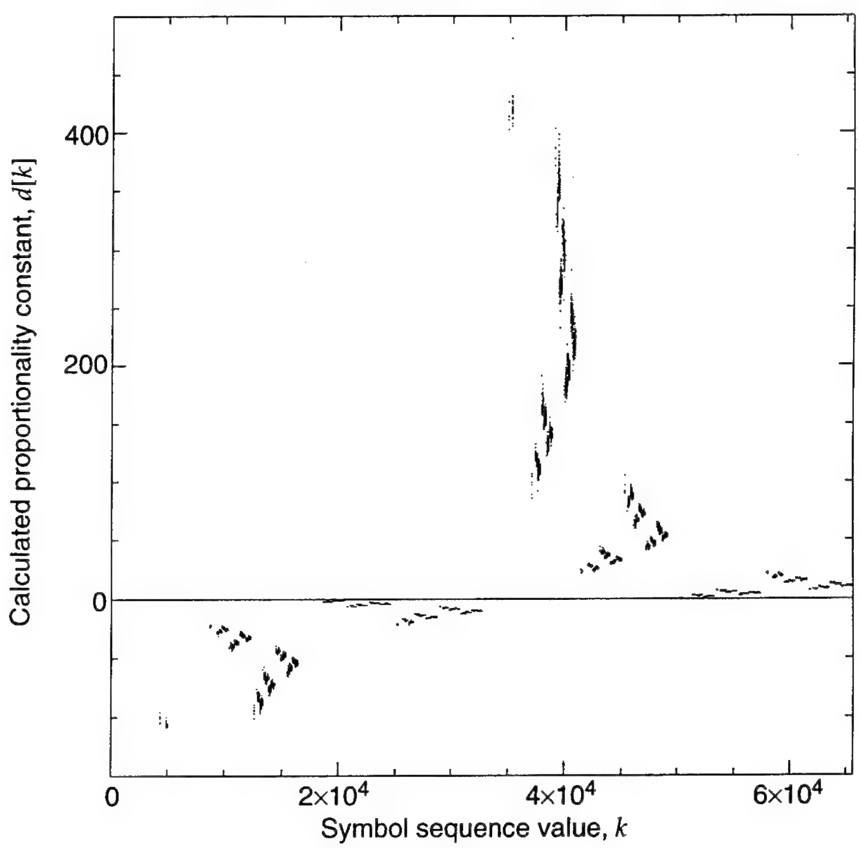
$$d[k] = \frac{p_{ref}}{\delta x_k} \ . \tag{12}$$

Figure 9 shows a plot of the calculated proportionality constant as a function of the symbol sequence k. (Like q, d has fractal structure.) If we define the initial sequence, or *source* sequence, $k_S = k^0$ and the *target* sequence, $k_T = k^{15}$, then the goal is to go from k^0 to k^{15} in 16 iterations. We make incremental steps by bit-wise shifting k^{15} into k^0: $k^0 \to k^1 \to k^2 \cdots \to k^{15}$. The bits from k^{15} are shifted into k^0, forming each subsequent symbol sequence. Control perturbations are calculated and applied so that a given state point sources a desired symbol sequence, rather than the symbol sequence it would normally have sourced under zero perturbation. Now we determine the actual state point that has occurred via iteration of the map, and compare it with the value of the inverse coding function that will source the desired symbol. The error in that comparison is used to calculate the control perturbation:

$$p_i = d\left[k^i\right]\left(x_i - s\left[k^i\right]\right) = \frac{x_i - s\left[k^i\right]}{\delta_{k^i}} p_{ref} \ . \tag{13}$$

The targeting examples in section 5 clearly demonstrate this.

Figure 9. Proportionality constant $d[k]$.



5. Targeting the Logistic Map

In targeting, we are trying to find an easy *route* from one state point to another. An optimal route would minimize the iterations needed to get from the source to the target, as well as minimize the control *energy* needed to travel there. We must point out that several routes will be possible in an ergodic system. If we wait long enough, the natural dynamics of the system may eventually carry a source point near a target point. However, there may be good reasons not to wait that long [11]. Targeting by controlling symbolic dynamics provides a navigational method. The number of iterations needed to get to within a *neighborhood* of the target and the perturbations needed to guide the trajectories there are completely determined by the number of bits in the symbol sequence register. Once we realize that a point in state space is uniquely linked to a symbol sequence, targeting becomes a simple matter of shifting in a target sequence (k_T) from a source sequence (k_S). The procedure is as follows:

- Iterate the system without and with perturbation to obtain $s[k]$ and $q[k]$.
- Define a source point x_S.
- Search $s[k]$ in order to find the closest value to x_S: $s[k_S] \cong x_S$.
- Define a target point x_T.
- Search $s[k]$ in order to find the closest value to x_T: $s[k_T] \cong x_T$.
- Sequentially shift in the bits from k_T into k_S. Calculate and apply the necessary perturbations for each successive shift by equation (13).

The first example is targeting an arbitrary state point from an arbitrary source point $x_S \to x_T = \tilde{x}_S \to x_T$. In this example, we demonstrate control of a period-16 orbit by simply targeting. Table 2 shows the results.

Source symbol $k_S = 28574 = D[0110111110011110]$ and target point $k_T = 56466 = D[1101110010010010]$, where $D[\ ]$ takes the decimal value of the quantity in the square brackets. Figure 10 shows the path taken on the attractor graphically.

A second example is what we would term a *rapid period-1 stabilization.* From an arbitrary source point, we target the fixed point ($k_T = 65535 = D[1111111111111111]$) and then repeatedly target the fixed point. Table 3 shows the results of this example.

In both examples, the control was carried out in the presence of uniformly distributed noise of 10^{-12} peak-to-peak amplitude and zero mean. The magnitude of the noise has a large effect upon the size of the perturbations needed to target and stabilize the system. Since the range of the map is approximately unity, the size of the perturbations shown in table 3 is good for general comparison. Figure 11 shows the results graphically.

Table 2. Arbitrary state-point targeting and demonstration of a period-16 control sequence.

Iteration i	Current symbol k	Associated state point* $s[k]$	Actual state point, $x(n+1)$	Difference between state and target points, $\lvert x(n+1) - x_T \rvert$	Control bit, b_i	Applied perturbation, p_i
Source, x_{S_1}	28574	1.966511E–01	1.966377E–01	4.388792E–01	—	—
0	57149	6.2715820341E–01	6.2714610232E–01	8.3707726855E–03	1	–5.2863171234E–05
1	48763	9.2830466935E–01	9.2830809745E–01	2.9279122244E–01	1	1.7791012268E–04
2	31990	2.6425563888E–01	2.6422396978E–01	3.7129290523E–01	0	3.0456946497E–04
3	63981	7.7184712831E–01	7.7186557569E–01	1.3634870068E–01	1	2.2956528329E–04
4	62427	6.9910684008E–01	6.9911418577E–01	6.3597310759E–02	1	5.5389791500E–05
5	59319	8.3509921281E–01	8.3511520937E–01	1.9959833436E–01	1	3.1054768026E–04
6	53102	5.4673072082E–01	5.4670301369E–01	8.8813861317E–02	0	–4.4358120314E–05
7	40668	9.8382616628E–01	9.8382975647E–01	3.4831288146E–01	0	8.7426558587E–04
8	15801	6.3167764857E–02	6.3171712545E–02	5.7234516247E–01	1	–2.3937827438E–04
9	31602	2.3492524516E–01	2.3493459110E–01	4.0058228391E–01	0	–1.0963326411E–04
10	63204	7.1352847336E–01	7.1354940069E–01	7.8032525677E–02	0	1.7461076372E–04
11	60873	8.1148216065E–01	8.1149040418E–01	1.7597352917E–01	1	1.3262176675E–04
12	56210	6.0733884086E–01	6.0732598829E–01	2.8190886722E–02	0	–4.6614076940E–05
13	46884	9.4674932521E–01	9.4675897847E–01	3.1124210346E–01	0	6.8965761921E–04
14	28233	2.0014128684E–01	2.0014823141E–01	4.3536864360E–01	1	–1.0344111909E–04
15	56466	6.3551220592E–01	6.3553644028E–01	1.9565267675E–05	0	1.1374004902E–04
$(x_T = \tilde{x}_S)$						
16	47397	9.1958692689E–01	9.1959694288E–01	2.8408006787E–01	1	4.4703382471E–04
17	29259	2.9354324144E–01	2.9356852297E–01	3.4194835204E–01	1	–2.0087323557E–04
18	58518	8.2326609079E–01	8.2328094156E–01	1.8776406655E–01	0	2.6642402486E–04
19	51501	5.7761373061E–01	5.7763181026E–01	5.7885064750E–02	1	4.6868864940E–05
20	37467	9.6857807135E–01	9.6858544389E–01	3.3306856888E–01	1	9.0439706514E–04
21	9399	1.2081458035E–01	1.2082541540E–01	5.1469145961E–01	1	–3.0801158981E–04
22	18798	4.2166424384E–01	4.2168701957E–01	2.1382985544E–01	0	–5.9718902906E–05
23	37596	9.6813298861E–01	9.6813773265E–01	3.3262085764E–01	0	5.7208161307E–04
24	9657	1.2247435751E–01	1.2248048826E–01	5.1303638675E–01	1	–1.6976738140E–04
25	19314	4.2664417153E–01	4.2667345610E–01	2.0884341891E–01	0	–7.1813376103E–05
26	38628	9.7113015096E–01	9.7113660814E–01	3.3561973313E–01	0	8.6391995156E–04
27	11721	1.1129923533E–01	1.1130449298E–01	5.2421238203E–01	1	–1.6571221563E–04
28	23442	3.9265628538E–01	3.9267934570E–01	2.4283752931E–01	0	–8.3757283727E–05
29	46884	9.4674932521E–01	9.4675466576E–01	3.1123779075E–01	0	3.8154485518E–04
30	28233	2.0014128684E–01	2.0014800027E–01	4.3536887474E–01	1	–9.9998254070E–05
31	56466	6.3551220592E–01	6.3553644116E–01	1.9566145705E–05	0	1.1374416990E–04

* Stored in unperturbed coding function.

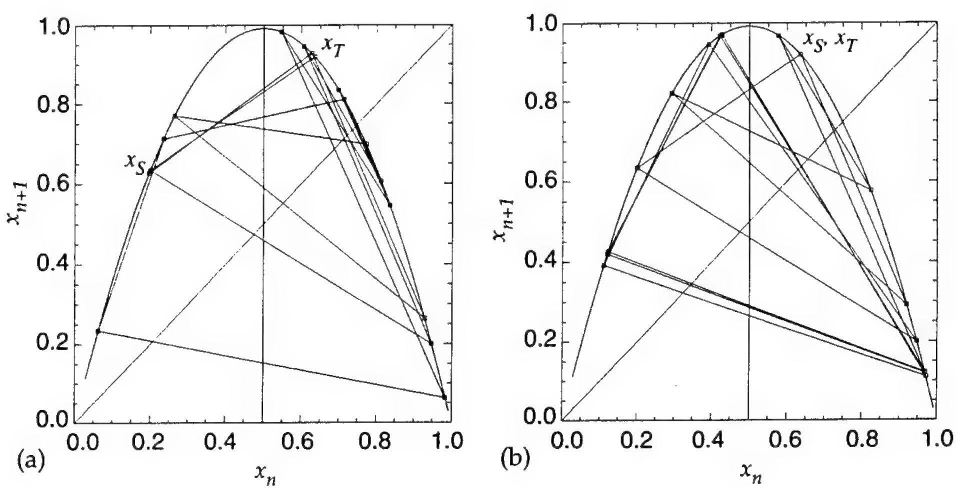
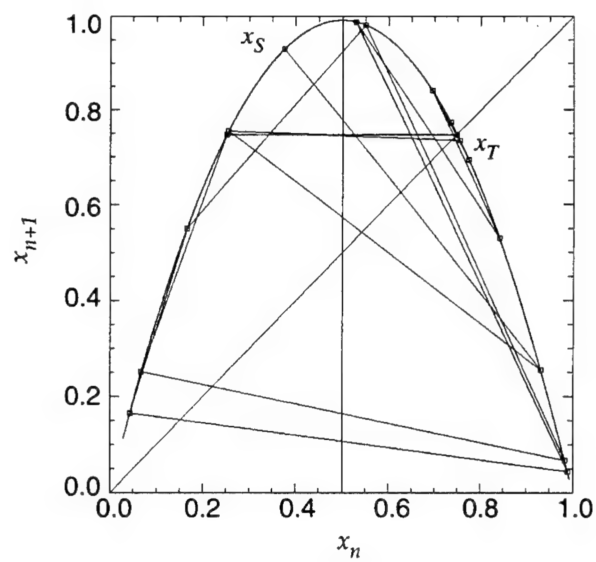


Figure 10. Graphical description of (a) targeting an arbitrary state point from an arbitrary source state point, and (b) a period-16 orbit.

Table 3. An example of rapid fixed-point stabilization.

Iteration i	Current symbol k	Associated state point* $s[k]$	Actual state point, $x(n+1)$	Difference between state and target points, $\lvert x(n+1) - x_T \rvert$	Control bit, b_i	Applied perturbation, p_i
Source, x_S	24524	3.754551E–01	3.754377E–01	3.726731E–01	—	—
0	49049	9.3092592112E–01	9.3090239932E–01	1.8279157431E–01	1	–1.2868674989E–03
1	32563	2.5531107999E–01	2.5528002008E–01	4.9283080493E–01	1	3.1896364866E–04
2	65127	7.5477741050E–01	7.5480580070E–01	6.6949756866E–03	1	3.1421345006E–04
3	64719	7.3478575310E–01	7.3480194841E–01	1.3308876602E–02	1	1.5615314549E–04
4	63903	7.7363884542E–01	7.7365656806E–01	2.5545743052E–02	1	2.2219717700E–04
5	62271	6.9520811550E–01	6.9523387801E–01	5.2876946997E–02	1	1.8898253249E–04
6	59007	8.4119409510E–01	8.4121846190E–01	9.3107636892E–02	1	4.9697939082E–04
7	52479	5.3031158533E–01	5.3033912777E–01	2.1777169724E–01	1	3.0639647084E–05
8	39423	9.8884884466E–01	9.8885339495E–01	2.4074256994E–01	1	1.6325515489E–03
9	13311	4.3769333556E–02	4.3776756851E–02	7.0433406816E–01	1	–6.5488648724E–04
10	26623	1.6617644513E–01	1.6615818535E–01	5.8195263966E–01	1	3.5019205078E–04
11	53247	5.5006584947E–01	5.5009060092E–01	1.9802022409E–01	1	4.2194143223E–05
12	40959	9.8254512851E–01	9.8254944157E–01	2.3443861656E–01	1	9.6400954124E–04
13	16383	6.8080295032E–02	6.8086293661E–02	6.8002453135E–01	1	–3.2305512045E–04
14	32767	2.5189261197E–01	2.5187818663E–01	4.9623263838E–01	1	1.5135440551E–04
15 = x_F	65535	7.4810942348E–01	7.4811771656E–01	6.8915503201E–06	1	8.7088069641E–05
16	65535	7.4810942348E–01	7.4811367768E–01	2.8526694139E–06	1	4.4674574484E–05
17	65535	7.4810942348E–01	7.4811364222E–01	2.8172131984E–06	1	4.4302238171E–05
18	65535	7.4810942348E–01	7.4811364191E–01	2.8169006118E–06	1	4.4298955605E–05
19	65535	7.4810942348E–01	7.4811364191E–01	2.8168978559E–06	1	4.4298926665E–05
			⋮			
31	65535	7.4810942348E–01	7.4811364191E–01	2.8168978313E–06	1	4.4298926407E–05

Figure 11. Graphical description of rapid fixed-point stabilization.



6. Constrained Sequences

If we consider targeting as equivalent to navigating routes from an origin to a destination, then symbolic dynamics provides a formalized road map for the journey. As is evident by figure 7, there are several symbol sequences that have either a very low or zero probability of occurrence. These *constraints* in the symbol space arise because the map, for $0 < r < 4$, does not extend over the whole range [0, 1] (see fig. 6). In fact, from equation (4) we determine that $x_{min} = 0.029551687512$, and thus the *minimal sequence* derived by iterating the map forward and collecting the symbols is B_{min} = 0001011001001011. The minimal sequence is the smallest sequence in magnitude, in a Gray counting order, possible by the map. Table 4 demonstrates this iteration.

Had x_{min} fallen exactly on one of the refinement boundaries (see fig. 6), we would have had a simple zero run-length constraint (for a run-length constraint on zeroes alone, no more than K zeroes can occur in a row). The state points would have mapped to peak in K iterations. Our constraint for $r = 3.97$ is much more complicated than this. However, by viewing the minimal sequence, we do know that no symbol sequence can occur that has more than three zeroes in a row within it.

A measure of the chaos of the map can be quantified in a measure called the topological entropy [18], h_T. In general, this entropy is an information-theoretical quantity that provides a measure of the possible states that a system can take on. In the context of chaotic dynamics, we can use this measure to quantify the number of orbits possible in a chaotic system. Since orbits in state space are linked to symbol sequences in this symbolic dynamics formalism, we can approximate this entropy by comparing the number of N-bit symbol sequences generated by the system versus all possible N-bit symbol sequences. This measure has a positive value for chaotic

Table 4. Iteration of x_{min} yielding minimal sequence B_{min} = 0001011001001011 (5707).

Iteration n	Point x_n	Corresponding symbol, b_n
0	0.0295516875 12	0
1	0.113853189575	0
2	0.400535843758	0
3	0.953224320043	1
4	0.177013231413	0
5	0.578347802852	1
6	0.968130638499	1
7	0.122489210033	0
8	0.426717845731	0
9	0.971180011687	1
10	0.111117904449	0
11	0.392119741566	0
12	0.946296543866	1
13	0.201752997885	0
14	0.639363441146	1
15	0.915393990149	1

systems and a value of zero for nonchaotic systems [4]. For the characterization that we carried out with one million iterations of the map, approximately 40 percent of the possible 65,536 sequences were allowed. The measure h_T is only approximated for a finite-length symbolic dynamics representation. It is calculated as

$$h_T = \lim_{N \to \infty} \frac{\log_2 n}{N} \cong \frac{\log_2 n}{16} \cong 0.92 \ , \tag{14}$$

where n = the number of allowed sequences and N = the number of bits. Other disallowed sequences were due to more complex symbol constraints and possibly occurred because of sequences that have a low probability of occurrence.

The constraint on symbol sequences presents two problems: (1) some regions may not be able to be targeted simply because they are not on the attractor and, most importantly, (2) derived routes may not be passable because they involve shifting in disallowed sequences. The second case was not at all uncommon. However, because the attractor exists, any existing state point on the attractor is accessible to any other existing state point. The route may be longer, but a route does exist.

One method of dealing with problem 2 is the insertion of a buffer sequence of length N_B between the source sequence and the target sequence (see fig. 12). The buffer sequence may or may not be as long as the source and target sequences. For example, if there is only a simple run-length constraint (i.e., any sequence containing, say, four zeroes or more in a row is disallowed), then inserting the one-bit buffer-sequence "1" will ensure that runs of more than three zeroes will not be generated by concatenation. If $N_B = N$, we call this buffer sequence a *jump point*, because it directly translates to a point in state space. It may be possible, for some systems, to find a single jump point in state space that will act as a conduit to and from all possible state points (i.e., the fixed point). If a single jump point does not exist, it may be possible to create a table of buffer words that will allow safe passage from varied source/target combinations. Table 5 shows an example of an attempt to target via an unpassable route (road block), and table 6 shows the insertion of a jump-point sequence to allow targeting to occur. Instead of 16 iterations, there are 32. The following bit shifting diagram shows a configuration in which a disallowed sequence of four zeroes would occur.

1111111111111100	$\leftarrow$	0011101110010010
Source sequence		Target

Figure 12. Sequence shifting diagram showing insertion of a buffer word.

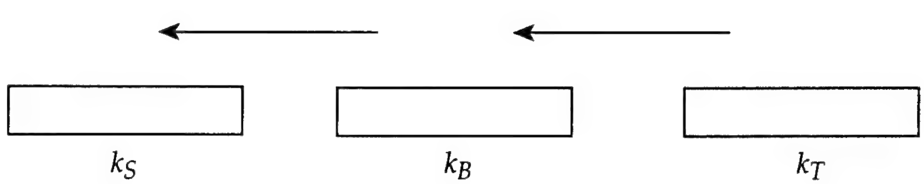


Table 5. Attempt to target via an unpassable route (roadblock): target = 7.627758E–02.

Iteration i	Current symbol k	Associated state point* $s[k]$	Actual state point, $x(n+1)$	Difference between state and target points, $\lvert x(n+1) - x_T \rvert$	Control bit, b_i	Applied perturbation, p_i
Source, x_S	65532	7.480572E–01	7.48572E–01	6.1718012E–01	—	—
0	65528	7.4823513083E–01	7.4821648997E–01	6.7193890867E–01	0	–1.9637686395E–04
1	65520	–1.0000000000E–03	7.4786570304E–01	6.7158812173E–01	0	0.0000000000E+00
2	65505	–1.0000000000E–03	7.4859355423E–01	6.7231597292E–01	1	0.0000000000E+00
3	65475	–1.0000000000E–03	7.4715900084E–01	6.7088141953E–01	1	0.0000000000E+00
4	65415	–1.0000000000E–03	7.4998239936E–01	6.7370481806E–01	1	0.0000000000E+00
5	65294	–1.0000000000E–03	7.4440999502E–01	6.6813241372E–01	0	0.0000000000E+00
6	65053	–1.0000000000E–03	7.5534716369E–01	6.7906958239E–01	1	0.0000000000E+00
7	64571	–1.0000000000E–03	7.3364742818E–01	6.5736984688E–01	1	0.0000000000E+00
8	63607	–1.0000000000E–03	7.7577330983E–01	6.9949572853E–01	1	0.0000000000E+00
9	61678	–1.0000000000E–03	6.9057791289E–01	6.1430033159E–01	0	0.0000000000E+00
10	57820	–1.0000000000E–03	8.4830989369E–01	7.7203231239E–01	0	0.0000000000E+00
11	50105	–1.0000000000E–03	5.1086052428E–01	4.3458294298E–01	1	0.0000000000E+00
12	34674	–1.0000000000E–03	9.9203179357E–01	9.1575421227E–01	0	0.0000000000E+00
13	3812	–1.0000000000E–03	3.1381774037E–02	4.4895807264E–02	0	0.0000000000E+00
14	7625	–1.0000000000E–03	1.2067598342E–01	4.4398402120E–02	1	0.0000000000E+00
15	15250	7.6277581300E–02	4.2126982206E–01	3.4499224076E–01	0	–1.6979509865E+01

Table 6. Insertion of a jump-point sequence: target = 7.627758E–02.

Iteration i	Current symbol k	Associated state point* $s[k]$	Actual state point, $x(n+1)$	Difference between state and target points, $\lvert x(n+1) - x_T \rvert$	Control bit, b_i	Applied perturbation, p_i
Source, x_S	65532	7.480572E–01	7.480572E–01	6.7178012E–01	—	—
0	65529	7.4821534803E–01	7.4821648997E–01	6.7193890867E–01	1	1.2020856267E–05
1	65523	7.4789678410E–01	7.4790496279E–01	6.7162738148E–01	1	8.5705721635E–05
2	65511	7.4851148670E–01	7.4853244204E–01	6.7225486074E–01	1	2.1938233599E–04
3	65487	7.4730639913E–01	7.4732090595E–01	6.7104332464E–01	1	1.5194295363E–04
4	65439	7.4968037285E–01	7.4969325758E–01	6.7341567628E–01	1	1.3654124576E–04
5	65343	7.4498850371E–01	7.4500919156E–01	6.6873161026E–01	1	2.1259724099E–04
6	65151	7.5420408066E–01	7.5422331552E–01	6.7794573421E–01	1	2.1137871668E–04
7	64767	7.3594964817E–01	7.3596015059E–01	6.5968256928E–01	1	1.0186318458E–04
8	63999	7.7146471177E–01	7.7148139845E–01	6.9520381715E–01	1	2.0284115455E–04
9	62463	6.9991503755E–01	6.9993828512E–01	6.2366070382E–01	1	1.7779241532E–04
10	59391	8.3382614509E–01	8.3383538790E–01	7.5755780659E–01	1	1.7822670304E–04
11	53247	5.5006584947E–01	5.5008387014E–01	4.7380628884E–01	1	3.0720087778E–05
12	40959	9.8254512851E–01	9.8254933758E–01	9.0627175628E–01	1	9.4076647475E–04
13	16383	6.8080295032E–02	6.8086352596E–02	8.1912287050E–03	1	–3.2622907802E–04
14	32767	2.5189261197E–01	2.5187824627E–01	1.7560066497E–01	1	1.5072867990E–04
15	65535	7.4810942348E–01	7.4811777508E–01	6.7184019377E–01	1	8.7702556859E–05
(jump point)						
16	65534	7.4812887863E–01	7.4811373712E–01	6.7183615582E–01	0	–1.5891211003E–04
17	65532	7.4805721716E–01	7.4807522026E–01	6.7179763896E–01	0	1.8914202384E–04
18	65529	7.4821534803E–01	7.4821668426E–01	6.7193910296E–01	1	1.4066086278E–05
19	65523	7.4789678410E–01	7.4790496517E–01	6.7162738387E–01	1	8.5730697460E–05
20	65511	7.4851148670E–01	7.4853244205E–01	6.7225486075E–01	1	2.1938251974E–04
21	65486	7.4733673283E–01	7.4732090595E–01	6.7104332464E–01	0	–1.6619514188E–04
22	65437	7.4964850806E–01	7.4963318281E–01	6.7335560151E–01	1	–1.6253677605E–04
23	65339	7.4508423833E–01	7.4507215149E–01	6.6879457019E–01	1	–1.2437608627E–04
24	65143	7.5405302280E–01	7.5403680808E–01	6.7775922678E–01	1	–1.7810303433E–04
25	64750	7.3625517555E–01	7.3626426862E–01	6.5998668732E–01	0	8.7520937169E–05
26	63964	7.7091680690E–01	7.7090845937E–01	6.9463087807E–01	0	–1.0179477427E–04
27	62393	7.0112712755E–01	7.0111824952E–01	6.2484066822E–01	1	–6.8284691909E–05
28	59250	8.3189343770E–01	8.3190500518E–01	7.5562742388E–01	0	2.1789570484E–04
29	52964	5.5521428967E–01	5.5519162743E–01	4.7891404613E–01	0	–4.2258818703E–05
30	40393	9.8039905049E–01	9.8039654353E–01	9.0411896223E–01	1	–4.9230768205E–04
31	15250	7.6277581300E–02	7.6290666287E–02	1.3084986331E–05	0	–6.4400482124E–04
(x_T)						

7. Two-Dimensional Maps

More general three-dimensional systems can yield more complicated two-dimensional Poincaré maps and thus two-dimensional return maps. An example of a two-dimensional map is generated by iteration of the expression

$$\left.\begin{aligned} x_{n+1} &= A - x_n^2 + By_n \\ y_{n+1} &= x_n \end{aligned}\right\} . \tag{15}$$

This map is called the Henon map [4]. Figure 13 shows a plot of the *strange attractor* (or fractal attracting set) that results when $A = 1.4$ and $B = 0.3$. Proper characterization of this system for symbolic dynamics targeting depends upon determining the correct generating partition. It has been shown that the proper generating partition exists where the *Kolmogorov entropy* (or metric entropy, h_U) is maximized [5]. This is a particularly difficult measure to calculate, especially with experimental data. In special cases where the metric entropy and the topological entropy are equal, finding the point(s) where h_T is maximized will produce the generating partition. Figure 14 shows the result of an approximation of the generating partition for the Henon map where this approach was used with 16-bit symbolic sequences.

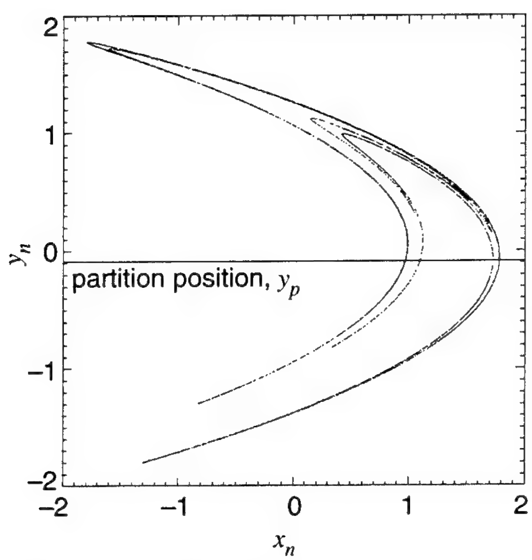


Figure 13. Strange attractor for two-dimensional Henon map (A = 1.4, B = 0.3).

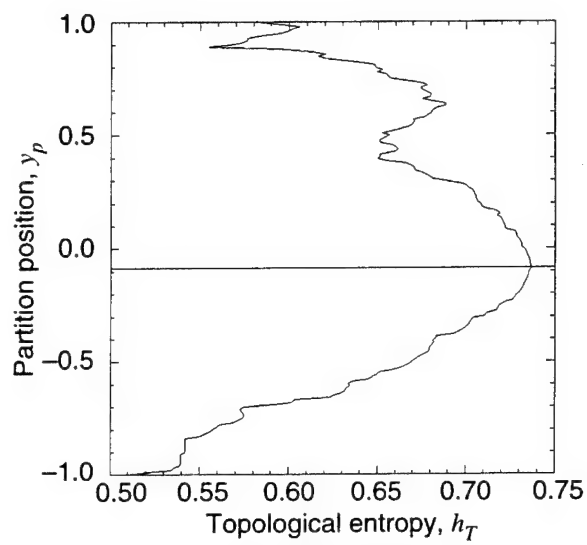


Figure 14. Topological entropy (h_T) approximation versus horizontal partition position y_p (showing peak occurring at y_p = −0.086).

8. Control Through Crisis: Ternary Systems

The logistic map undergoes a *crisis* when $\rho > 4$. In this case a region defined by

$$x_{c_1}, x_{c_2} = \frac{1}{2}\left(1 \mp \sqrt{1 - \frac{4}{\rho}}\right) \tag{16}$$

will project the map to a point greater than 1, and consequently send it off to $-\infty$ (see fig. 15). If it is possible to characterize the system near, but not at, crisis, then it is possible to form coding functions based upon a ternary symbol structure. For example, "0" is assigned to any point passing to the left of, "1" to any point passing to the right of, and "2" to any point passing through the region (this method would demand that the maximum crisis region be known a priori). Simply stated, any region of state space that leads to crisis can be given the symbol "2." If the system is targeted, or controlled, through only the sequences containing 0's and 1's (thus binary), then even when $\rho > 4$, the trajectories will never enter the crisis region.

For control to be maintained during crisis, the near-crisis characterization must be extrapolated. This brings to bear the notion of controlling a chaotic system under dynamic parameter conditions. For a system where the description of the dynamics is known explicitly, such as the logistic map, it would be possible to project the variations of the symbolic sequences due to varying ρ. Unperturbed and perturbed coding functions can be determined for various values of ρ. For systems where there is no explicit analytical definition of the map, it may be possible to curve-fit and parametrize the map. Rapid characterization could then be calculated by the method described in section 3. A single-humped map with a positive first derivative in the "0" partition will have a Gray ordering. Regions other than those that induce crisis can be disallowed by judicious choice of the ternary partitions.

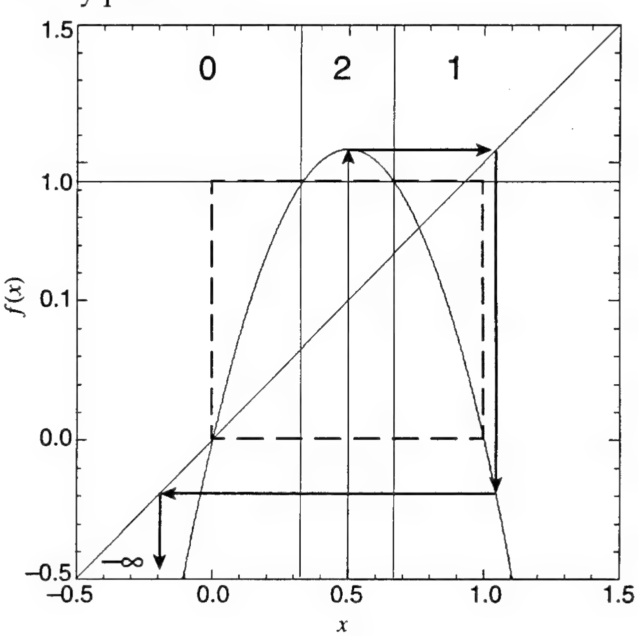


Figure 15. Logistic map undergoing crisis for $\rho > 4$.

9. Conclusion

We have shown that the formalism developed to control symbolic dynamics in chaotic systems using small perturbations readily lends itself to targeting. An important application that could be affected by this characteristic is rapid stabilization of collapsible systems. Studies of voltage collapse models for power-delivery networks have shown that chaotic oscillations often precede collapse [11]. Systems of this type are undergoing a form of crisis. An algorithm of this type can be applied to quickly guide the trajectories of these systems to regions of state space where periodic stabilization, using the OGY principle [3], can be performed.

This targeting method readily lends itself to digital processing techniques. The coding functions themselves are arrays whose storage space depends largely upon the number of bits defined for the symbol sequences. Once characterization is complete, only $s[k]$ and $d[k]$ need be stored, which are double-precision arrays 2^N long, where N is the number of bits in a symbol sequence. One could even quantize the state points in order to deal strictly with integer quantities. This further reduces needed storage space and speeds up calculation time. For each iteration, the control perturbations are calculated by a subtraction and multiplication (see eq (13)). Given the proper generating partition, characterization and targeting for two-dimensional maps will proceed in the same manner as for a one-dimensional map.

Although we know the analytical expression for the logistic map, it was not *necessary* to know it in order to target the system. A long enough period of observation and learning is sufficient to create a *model from measurement*. This is the approach that many researchers and experimenters continue to have success with.

References

1. Francis Moon, *Chaotic Vibrations*, Wiley & Sons, New York (1987).
2. Martin J. Hasler, *Electrical Circuits with Chaotic Behavior*, Proc. IEEE **75**, No. 8 (August 1987).
3. E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990).
4. Edward Ott, *Chaos in Dynamical Systems*, Cambridge Univ. Press, Canada (1993).
5. J. P. Crutchfield and N. H. Packard, *Symbolic Dynamics of One-Dimensional Maps: Entropies, Finite Precision, and Noise*, Int. J. Theor. Phys. **21**, 433–466 (1982).
6. S. Hayes, C. Grebogi, and E. Ott, Phys. Rev. Lett. **70**, 3031 (1993).
7. S. Hayes, C. Grebogi, E. Ott, and A. Mark, Phys. Rev. Lett. **73**, 1781 (1994).
8. C. M. Glenn and S. Hayes, *Observation of Chaos in a Microwave Limiter Circuit*, IEEE Microwave Guided Wave Lett. **4**, No. 12 (December 1994).
9. S. Basu, S. A. Mass, and T. Itoh, *Quasi-Periodic Route to Chaos in a Microwave Doubler*, IEEE Microwave Guided Wave Lett. **5**, No. 7 (July 1995).
10. Rajarshi Roy, T. W. Murphy, Jr., T. D. Maier, Z. Gills, and E. R. Hunt, *Dynamical Control of a Chaotic Laser: Experimental Stabilization of a Globally Coupled System*, Phys. Rev. Lett. **68**, 1259 (1992).
11. H. Chang, I. Dobson, R. Thomas, J. Thorp, and L. Fekih-Ahmed, *On Voltage Collapse in Electric Power Systems*, IEEE Trans. Power Syst. **5**, No. 2, 601 (May 1990).
12. T. Shinbrot, E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **65**, 3215 (1990).
13. Edward N. Lorenz, *Deterministic Nonperiodic Flow*, J. Atmos. Sci. **20** (March 1963).
14. Chai Wah Wu and Nikolai F. Rul'kov, *Studying Chaos via 1-D Maps—A Tutorial*, IEEE Trans. Circuits Syst. **40**, No. 10 (October 1993).
15. E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **38**, 3688 (1988).
16. A. N. Kolmogorov, Doklady Akademii Nauk **119**, 861 (1958).
17. E. N. Gilbert, *Gray Codes and Paths on the n-cube*, Bell Syst. Tech. J. **37**, 815–826 (1958).
18. R. C. Adler, A. C. Konheim, and M. H. McAndrew, *Topological Entropy*, Trans. Am. Math. Soc. **114**, 309 (1965).

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Hqdtrs Dept of the Army
Attn DAMO-FDQ MAJ McGonagle
400 Army Pentagon
Washington DC 20310-0460

US Army Mis Cmnd
Attn AMSMI-RD-WS C Bowden
Redstone Arsenal AL 35898-5248

US Army Rsrch Ofc
Attn J Mink
PO Box 12211
Research Triangle Park NJ 27709-2211

Nav Rsrch Lab
Attn Code 5540 J Palmer
Attn Code 5540 S Lessin
4555 Overlook Ave SW
Washington DC 20375-5336

Clemson Univ
Dept of Electrl & Cmptr Sci
Attn C H Butler
Clemson SC 29634

Northwestern Univ
Lab for Chaos Dept of Chem Engrg
Attn T Shinbrot
Evanston IL 60208

Rensselaer Polytechnic Inst
Dept of Physics
Attn J Haus
Attn R Lichtenstein
Troy NY 12180-3590

The Johns Hopkins Univ
ECE Dept
Attn C R Westgate
Attn J Khurgin

The Johns Hopkins Univ (cont'd)
ECE Dept
Attn R Joseph
Attn J Sinsky
25 Charles Stret
Baltimore MD 21218

The Johns Hopkins Univ
Materials Science Dept
Attn J Spicer
Attn J Wagner
Attn D Oursler
25 Charles Stret
Baltimore MD 21218

The Johns Hopkins Univ
Applied Physics Lab
Attn J Sommerer
Attn J Franson
M S Eisenhower Rsrch Ctr
Laurel MD 21045

Univ of Maryland
Dept of Physics &
Astronomy
Attn D Currie
College Park MD 20742

Univ of Maryland Lab for Plasma Rsrch
Attn C Grebogi
Attn E Ot
College Park MD 20742

US Army Rsrch Lab
Attn AMSRL-IS-TG J Gowens
115 O'Keefe Bldg
Atlanta GA 30332-0862

US Army Rsrch Lab
Attn AMSRL-WT D Eccleshall
Aberdeen Proving Ground MD 21005-5425

US Army Rsrch Lab
Attn AMSRL-SL-CM B Ruth
Attn AMSRL-SL-CM J Soln
Edgewood MD 21010-5423

Distribution

US Army Rsrch Lab
Attn AMSRL-SE-RT B Weber
FT Belvoir VA 22060

US Army Rsrch Lab
Attn AMSRL-IS-TA B Sadler
Attn AMSRL-IS-TA C Glenn (100 copies)
Attn AMSRL-IS-TA M Smith
Attn AMSRL-IS-TA S Hayes
Attn AMSRL-LT R Gilbert
Attn AMSRL-OP-SD-TA Mail & Records Mgmt
Attn AMSRL-OP-SD-TL Tech Library (3 copies)
Attn AMSRL-OP-SD-TP Tech Pub (5 copies)
Attn AMSRL-PS-AA D Wortman
Attn AMSRL-PS-AA M Tobin
Attn AMSRL-PS-PD A Abou-Auf
Attn AMSRL-PS-PD J McGarrity
Attn AMSRL-SE J M Miller
Attn AMSRL-SE J Pellegrino

US Army Rsrch Lab (cont'd)
Attn AMSRL-SE-EA R Tobin
Attn AMSRL-SE-EO A Filipov
Attn AMSRL-SE-EO J Goff
Attn AMSRL-SE-EO N Gupta
Attn AMSRL-SE-ES M Patterson
Attn AMSRL-SE-R A Sindoris
Attn AMSRL-SE-RI D Rodkey
Attn AMSRL-SE-RU J Sichina
Attn AMSRL-SS V DeMonte
Attn AMSRL-SE-EP B Stann
Attn AMSRL-WT-NB H Brandt
Attn AMSRL-WT-NF C Fazi
Attn AMSRL-WT-NG A Bromborsky
Attn AMSRL-WT-NH G Huttlin
Attn AMSRL-WT-NH J Corrigan
Attn AMSRL-WT-NH L Libelo
Attn AMSRL-WT-NH M Litz
Attn AMSRL-WT-PC A Cohen
Attn AMSRL-WT-PC R Beyer
Adelphi MD 20783-1197